

前回の動画の 補足説明

一つ、前回の動画で 説明し忘れた事が ありました。それは、ESP32マイコンには **Deep Sleep**という省電力機能がある事です。

前回の様なロガーの場合は 一定周期でセンサーのデータを サンプリングして、トリガ等の条件が成立すれば SDカードに書き込みます。

このような処理の場合、一定周期の中で 次の周期が来るまで 大なり小なりの時間待ちが発生します。特にバッテリー駆動の場合時間待ちで 消費する電力を押さえる事でバッテリーの駆動時間を延ばす事が出来るはずです。

ロガーには、1分に1回とか長周期のデータを サンプリングする物もあれば、今回の 100Hzサンプリング等の 中速度のサンプリングレイトの物、そして 特殊用途では 5μ 秒で

サンプリングする物も あります。これは、ハード的な手法を使わないと スピードが 間に合いません。通常のマイコンで可能な守備範囲は、中速度ぐらいまでと考えます。

で、前回までのロガーは 100Hzサンプリングで 10ミリ秒周期でしたが、そのうち 約 1ミリ秒が、タイマー割り込み処理で、残りの処理が SDカードに書き込んで無ければ 1ミリ秒で終わると思います。そう考えると 小間切れでは ありますが 10ミリ秒の サンプルレイト内で 待ち時間は 8ミリ秒になります。4/5が 待ち時間という事ですね。という事であれば 単純計算で消費電力を 1/5に 減らせる事になります。

実動時間	待機時間

じゃ、何故 **DeepSleep**機能を 使用しなかったのかというと、実は 最初 DeepSleep機能を使えば、単三乾電池駆動も出来るのではと思つてました。

私は **ESP32と Arduinoの書籍**を参考書にしていますが、その本の DeepSleep機能の説明にて 一定周期で **DeepSleep** させる機能があります。 2本の専用の関数で制御します。

- ① **esp_sleep_enable_timer_wakeup**関数と
- ② **esp_deep_sleep_start**関数です。

esp_sleep_enable_timer_wakeup関数では、
DeepSleepしてから 復帰するまでの時間を
マイクロ秒単位で 指定します。 この関数を
実行した後に、**esp_deep_sleep_start**関数を
実行して DeepSleep状態に入ります。

で、指定時間経過後、復帰すると、電源ON、
CPUリセット時と同じく **setup**関数から呼ばれま
す。 **ESP32**の省電力機能は CPUとは独立した
ulp という 超低消費電力の コプロセッサが あ
ります。この **ulp**の側で タイムカウントして 時
間になつたら CPUが 叩き起こされる。 とい
う事だと思います。 その間は CPUは 完全に停止
しているようです。 よって CPUリセットと 同じ
シーケンスで起動する。 という事でしょう。

setup関数内で 単純に1サンプル データを取
り込んで SDカードに アpendモードで 書き込
む様な 单発的な使い方であれば、使えるでし
ょが、 前回までのロガーは リングバッファに
5秒、10秒前までのデータを 貯め込む処理をして
いるので、1/100秒周期のwakeupの度に メモ
リーも初期化されると **データを貯め込むリング
バッファを構築出来ない** という事だったのです。

AVRマイコンの 概要

では、今回のテーマに 入ります。

最初は、AVRマイコンの概要について 説明
しようかと思っていたら、過去に 「065 絶滅危
惧種 百円マイコン比較レビュー」にて、AVRマ
イコンの事も 紹介していた事を 思い出しまし
た。 自分も忘れていたので見てみました。

余分に、PICマイコンと R8Cマイコンも 説明
していますが、切り離しが難しい様なので、そ
のまま説明部分を この動画に 入れる事に
しました。

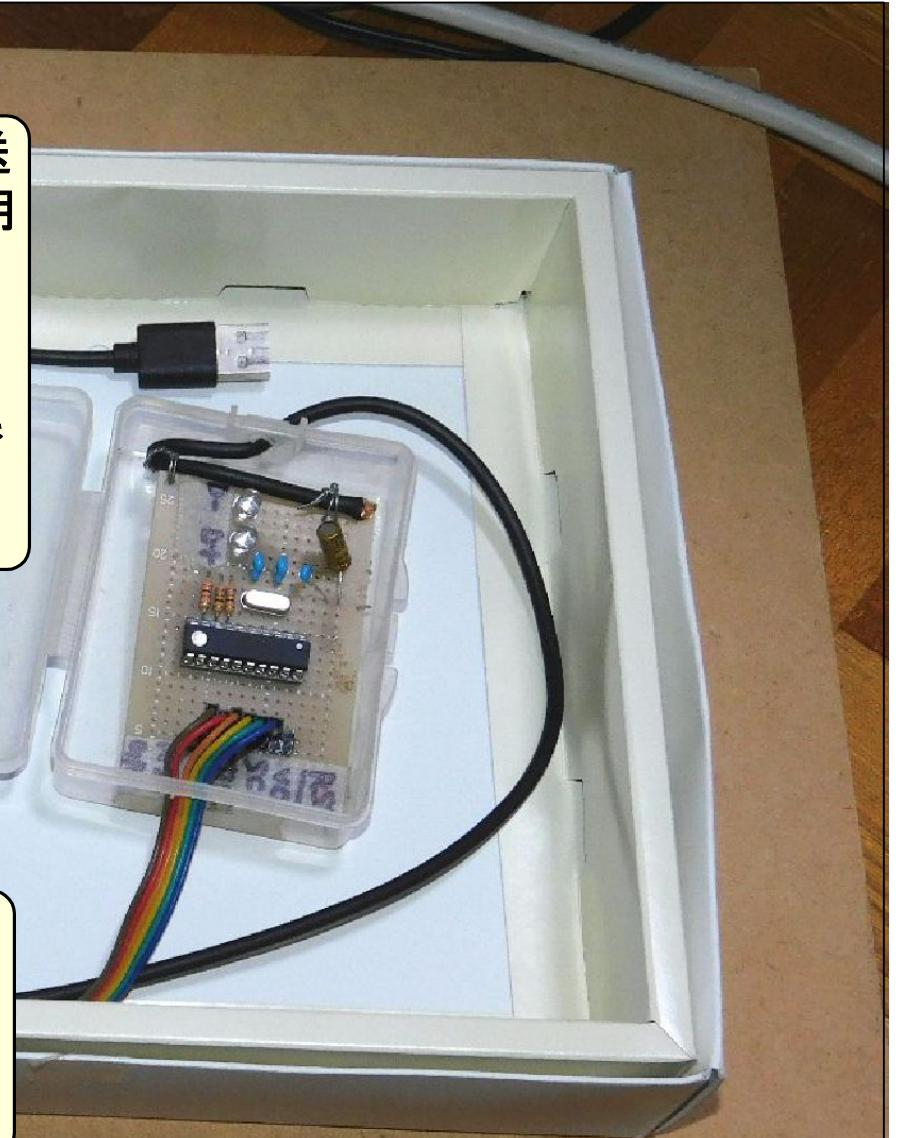
概要の説明は しっかりやってますので、つ
いでに、PICマイコンと R8Cマイコンの説明も
御試聴下さい。

AVRマイコン書き込みソフト hidspxとは

実は、2024年10月上旬、Uさんから この画像の物が 送られてきました。これは Uさんが自作したAVRマイコン用書き込み器 hidspxです。左の CD-Rに ソフトが 入っています。元々の設計、製作者は 山形県立産業技術短大のSさんという方です。Uさんは この書き込み器、及び書き込みソフトに思い入れがあるようで、良ければ、YouTubeで紹介してほしい。との事で 動画を作成する事にしました。



因みに 私と Uさんは どのように知り合ったのかというと、107の動画で、R8C/Mシリーズの小基板を やや多めに作成し、何人かの視聴者の方々に配ったのですが、そのうちの一人です。



資料を見直すと hidspxが 書き込み器の事で、発音も ハイドspxかもしれません。

Uさんは Linux環境で使われていたようなので、Linuxを 中古のノートPCに インストールする事にしました。

過去に Windows10が インストールされている デスクトップPCの HDDの後方に 空きパーティションを用意して Linux Ubuntuを インストールした事が ありました。OSの 切り替えは glbで やっていました。が、Windowsにアップデートが かかった時、何と glb が 自動的に 外されて Windowsは起動するけど、Linuxは 起動出来なくなる。という事が 発生しました。このような事があり しばらくは Linuxをインストールする事は やってなかつたのですが、元々 Windows7が インストールされていた中古ノートPCを 購入し 最初に

パーティションのバックアップを 取ろうとしたら Windowsの先頭部分が壊れたようで、インストールディスクで修復しようとしたが、中古PC屋の方で 何か変な事をしているようで修復できませんでした。しばらく放置していたのですが、230GBの SSDが 入ってましたが、別PCに 増設で 接続して SSD内のすべてのパーティションを 削除してノートPCに戻しました。

そして、昔のノートPCは、32bitOSでないと、インストール出来ない物が 多いです。

Linux も その大半が 64bitに移行し終わっていて、メジャーな ubuntuも全て 64bitに 変更されています。メニューに 32bitが残されてる場合もありますが、その先に行こうとすると **Not Found.** です。

で、見つけたのは、次のページで 紹介します。

32bit Linux ダウンロードサイト：
<https://github.com/peabee/releases/releases>
必要であれば、書き写して下さい。

2 weeks ago

github-actions

BookwormPup...

9e7b2da ✓

Compare ▾

BookwormPup32-23.12

Latest

BookwormPup32/release-241115

Update release-upupnn.yml back to k6.6

▼ Assets 8

BookwormPupは 本の虫の子犬でしょうか。 かわいい名前ですね。

紹介者のお勧めには
入ってませんが

先頭に出て来たの
で、ダウンロードして
みました。

ISOファイルをダウンロードして
CD-Rに 書き込んで下さい。
そして インストールする PCに
持って行って下さい。

因みに BookwormPup Linux は
このサイトの 32bit OSでは
566MB と ちょっと大きめです。

BookwormPup32-23.12-241115.iso	566 MB	2 weeks ago
devx_dpupbw32_23.12.sfs	103 MB	2 weeks ago
docx_dpupbw32_23.12.sfs	32.8 MB	2 weeks ago
kernel_sources-6.1.115-kernel-kit.sfs	173 MB	2 weeks ago
nlsx_dpupbw32_23.12.sfs	119 MB	2 weeks ago
SHA512checksums.txt	801 Bytes	2 weeks ago
Source code (zip)		3 weeks ago
Source code (tar.gz)		3 weeks ago

JammyPup32-22.04

JammyPup32/release-241101

Update release-upupnn.yml temp k6.1

ISOファイルをダウンロードして
CD-Rに書き込んで下さい。
そして インストールする PCに
持って行って下さい。

因みに JammyPup Linux は
このサイトの 32bit OSでは
385MB と コンパクトです。

▼ Assets

JammyPupは 扱いやすい子犬でしょうか。

devx_upupjj+d_22.04.sfs	152 MB	last month
docx_upupjj+d_22.04.sfs	31.9 MB	last month
JammyPup32-22.04-241101.iso	385 MB	last month
kernel_sources-5.15.170-kernel-kit.sfs	153 MB	last month
nlsx_upupjj+d_22.04.sfs	24.1 MB	last month
SHA512checksums.txt	799 Bytes	last month
Source code (zip)		last month
Source code (tar.gz)		last month

紹介者のお勧めです。

32bit Linusの中でもコンパクトです。
ダウンロードしてみました。

BionicPup32ng-19.03

BionicPup32/release-241101

Update release-upupnn.yml temp k6.1

ISOファイルをダウンロードして
CD-Rに 書き込んで下さい。
そして インストールする PCに
持つて行って下さい。

因みに BionicPup Linux は
こここのサイトの 32bit OSでは
368MB で、
今回選択した3つの中では 最小です。

▼ Assets

8

BionicPupは 生体工学な子犬でしょうか。

 BionicPup32-19.03-241101.iso	368 MB	29 days ago
 devx_upupbb_19.03.sfs	110 MB	29 days ago
 docx_upupbb_19.03.sfs	29.5 MB	29 days ago
 kernel_sources-5.15.170-kernel-kit.sfs	153 MB	29 days ago
 nlsx_upupbb_19.03.sfs	19.9 MB	29 days ago
 SHA512checksums.txt	794 Bytes	29 days ago
 Source code (zip)		29 days ago
 Source code (tar.gz)		29 days ago

紹介者のお勧めです。

32bit Linuxの中でもコンパクトです。
ダウンロードしてみました。

Linuxのインストールは どれでもいいのですが、今回は、最初に ダウンロードした BookwormPup Linux 32-23.12 Linux を 中古ノートPCに インストールしてみます。

今回の AVRマイコンの書き込み器の環境だけでなく、Linux上で 組み込み開発の環境が揃って来ているし、古くなったPCの再利用に Linuxの選択肢は 有効と思います。

ちなみに 私が Linux環境でやってみたいのは、ルネサスマイコンのコンパイラが Linux上で揃って来ているので RXマイコンの開発とか やってみたいのと、あと、Linux上の アプリ開発にて ラザルスを 使ってみたいという事があります。ラザルスは、言語が Object Pascal と言う事で、嫌がられる事も ありますが、Windows上で 長年 Delphiを やって来た 私にとっては 扱いやすい環境です。

それと、Windows上で Linuxのコマンド使える環境が ありますが マイクロソフトが 用意した環境は バージョン1の時は 良かつたらしいですが、バージョン2になって 今まで動いていた物が 動かなくなつた。、との事です。

この事からも リナックスを OSから入れた方が いいです。

Linux上で ある程度 環境が 整ってきたら そのあたりも YouTube動画にしてみたいな。と、思います。

一つの事やら

では、インストールの動画に 移ります。

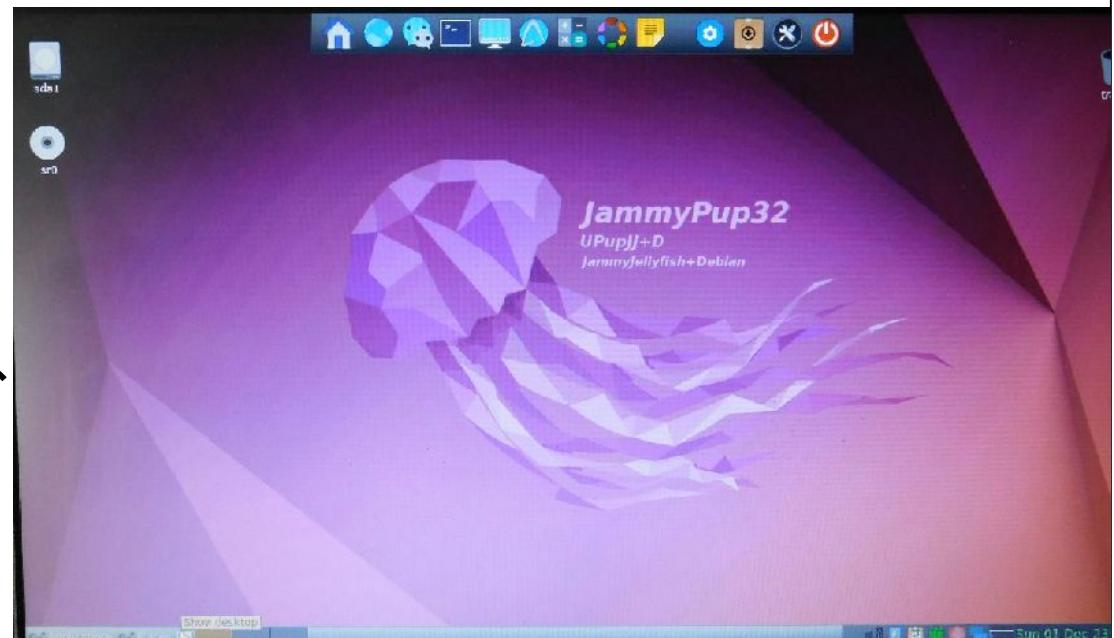
今回の Linuxインストールが 出来ませんでした。

最初に [BookwormPup Linux](#)の インストールを行いましたが、インストールの途中で、画面が真っ暗になって、だんまりになってしまいました。 という事で [BookwormPup Linux](#) のインストールは 出来ませんでした。

次に [JammyPup Linux](#)を インストールしましたが、一応 インストール作業が終わり X window が 起動しました。

上の [BookwormPup Linux](#) とインストール時に出てくるメッセージは 同じようなので ベースは 同じ物と思われます。 で、BookwormPup Linuxが 真っ暗になったタイミングは X windows を 表示し始めるタイミングだったようです。 よって [BookwormPup Linux](#) には、私の ノートPCの ディスプレイ インタフェースにマッチする ディスプレイドライバが 無かったの

では、ないかと思われます。 [JammyPup Linux](#) では、GUIの画面は 正常に出了ました。 ローポリの くらげのような壁紙が 出てきました。 見た目は、かっこいいです。



でも、その後 問題が発生しました。
再起動できません。

実は、Linuxのモジュールを ハードディスクではなくて、何と RAMに 転送しているのです。

```
*** JammyPup32 22.04 - Linux 5.15.170 [i686] ***
e2fsck 1.46.2 (28-Feb-2021)
/dev/sda1: clean, 133377/15630336 files, 2802211/62514432 blocks
Waiting 5 seconds for slow storage devices [0/5].
Waiting 5 seconds for slow storage devices [1/5].
Waiting 5 seconds for slow storage devices [2/5].
Waiting 5 seconds for slow storage devices [3/5].
Waiting 5 seconds for slow storage devices [4/5].
Copying 'puppy_upupjj+d_22.04.sfs' to ram...
Loading the 'puppy_upupjj+d_22.04.sfs' main file...
Copying 'adrv_upupjj+d_22.04.sfs' to ram...
Copying 'zdrv_upupjj+d_22.04.sfs' to ram...
Copying 'fdrv_upupjj+d_22.04.sfs' to ram...
```

で、電源OFFや、再起動を行うと メモリがゼロクリアされるので、RAM上の Linuxは 無くなります。どうやら、一旦 RAM上に Linuxを構築して USBメモリに コピーして USBメモリから起動できるらしいんですけど、そのコピーの操作を行い、BIOSの BOOTシーケンスをUSBデバイスを先頭に 持って来ても起動できませんでした。?

基本的に 今回ダウンロードした Linuxは 3つとも、最初 Linuxを RAM上に構築するようになっているようです。多分 ハードディスクに コピーする手段もあるのではと思いますが、やりかたが分かりませんでした。

それと、インストール時、ログインの IDや パスワードを設定する画面も出てきませんでした。いきなりGUI環境に入れます。

これって、セキュリティ的には よろしく無いと思うのですが、こういう仕様なのでしょうか。?

今回の Linuxは 時間が十分ある時に やることにして、今回は 別の 32bit Linuxのインストーラが 無いか探してたら、[5年前に書き込んだ32bitの Lubuntu](#) なるDVDがありました。ubuntuの軽量版との事です。これでインストールしたら普通に SSDに インストール出来ました。